# Tutorial: Servidor Soap utilizando Ruby

## Elaboración Servicio Web Soap

A continuación se detallan los pasos para el desarrollo de un Servicio Web Soap mediante Ruby.

## Herramientas a instalar

Para poder realizar servicios web utilizando Ruby, es necesario tener instalados las siguientes herramientas:

* Ruby
* Rails
* Savon
* Gem wash\_out

### Instalar Ruby en Windows

La instalación de Ruby en Windows se puede realizar de dos formas, utilizando Ruby Versión Manager (RVM) o un instalador (.exe).

En este tutorial se utilizará RubyInstaller porque la herramienta RVM dejó de tener soporte en 2012. Para descargar RubyInstaller hay que ir a la página <http://rubyinstaller.org/downloads/> y seleccionar el instalador Ruby 2.1.5 o Ruby 2.1.5(x64).

Para ver que Ruby ha sido instalado correctamente y que la versión es 2.5.1 es suficiente con escribir ruby –version en la línea de comandos.

ruby --version

### Instalar Ruby en UNIX

Para la instalación de Ruby en sistemas operativos UNIX se debe utilizar la herramienta Ruby Versión Manager (RVM). El comando para instalar RVM y la última versión estable de Ruby es:

\curl -L https://get.rvm.io | bash -s stable --ruby

### Instalar Rails (Windows y UNIX)

Rails es un framework de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby. Para instalar Rails en Windows o en cualquier sistema operativa UNIX, el comando es el mismo. El comando que se debe escribir en la línea de comandos es:

gem install rails

### Instalar Savon (Windows y UNIX)

Savon (<http://savonrb.com/>) es un cliente SOAP para Ruby creado por Daniel Harrington. La versión que se utiliza en este tutorial es la versión 2.9.0. Para instalar Savon en Windows o en cualquier sistema operativa UNIX el comando es el mismo. El comando que se debe escribir en la línea de comandos es:

gem install savon

### Problemas instalando la gema

Al intentar instalar gemas a veces se obtiene el siguiente problema “*Unable to download data from https://rubygems.org/ - SSL\_connect returned=1 errno=0 state=SSLv3 read server certificate B: certificate verify failed*”. Para solucionar este problema, se debe ejecutar en la línea de comandos:

sudo gem sources –r <https://rubygems.org>

sudo gem sources –a <http://rubygems.org>

## Creación del proyecto

1. Crear directorio “*TutorialRuby*” en C.
2. Abrir el cmd e ir al directorio donde se encuentra “*TutorialRuby*”.

cd C:\TutorialRuby

1. Crear un nuevo proyecto Rails utilizando el comando “*rails new nombreproyecto*”. En este caso el nombre del proyecto será “*SoapServer*”.

rails new SoapServer

1. Una vez terminada la creación del proyecto, deberá tener los siguientes directorios en “*C:\TutorialRuby\SoapServer*”.

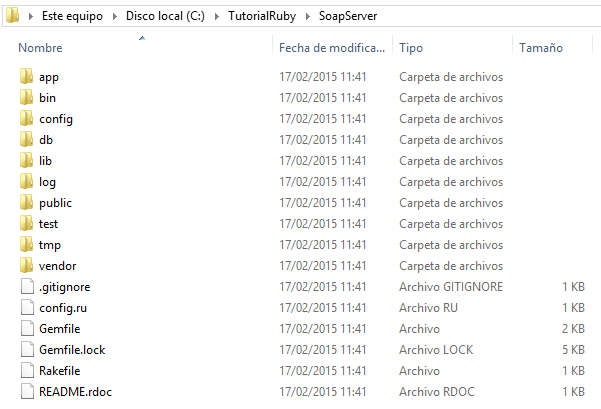


Ilustración 1: Directorios del proyecto SoapServer

## Implementación del Servicio Calculator

1. Buscar el fichero “*Gemfile*” dentro del directorio “*C:\TutorialRuby\SoapServer*”.
2. Buscar la línea “# gem 'therubyracer', platforms: :ruby”.
3. Añadir las siguientes líneas después de “# gem 'therubyracer', platforms: :ruby”.

# Gem to simplify the creation of SOAP service providers

gem 'wash\_out'

1. Acceder mediante la línea de comando al directorio “*C:\TutorialRuby\SoapServer*”.

cd SoapServer

1. Ejecutar “*bundle install*” en la línea de comandos para instalar el gem ‘wash\_out’.

bundle install

1. Para verificar que se ha instalado el gem correctamente, se debe buscar la línea “*Using wash\_out 0.9.2*” en la línea de comando.



Ilustración 2: Gem wash\_out instalado

1. Una vez instalado el gem, hay que acceder al directorio “*SoapServer/app/controllers*” mediante la línea de comandos.

cd C:\TutorialRuby\SoapServer\app\controllers

1. Crear el controlador calculator con el comando

rails generate controller calculator\_soap add

1. Abrir el fichero calculator\_soap\_controller con un editor de texto y copiar el código.

class CalculatorSoapController < ApplicationController

soap\_service namespace: 'urn:WashOut'

# Add

soap\_action "add",

:args => { :a => :double, :b => :double},

:return => :double

def add

render :soap => (params[:a] + params[:b])

end

end

1. Buscar el fichero “*routes*” dentro del directorio “*SoapServer/config*” y copiar al final del fichero la siguiente línea.

wash\_out :calculator\_soap

1. Acceder al directorio “*bin*” mediante línea de comando y arrancar el servidor.

cd C:\TutorialRuby\SoapServer\bin

rails server

1. Para verificar que todo funciona correctamente, copiar en el navegador la url “[*http://localhost:3000/calculator\_soap/wsdl*](http://localhost:3000/calculator_soap/wsdl)”.

## Creación del cliente en Ruby

1. Crear y acceder al directorio “*SoapClient*” dentro de “*TutorialRuby*”.
2. Abrir un editor de texto, copiar el código y guárdalo con el nombre “*soap\_client.rb*”. Revisar que no guarda con la extensión .txt.

require 'savon'

class Client

def initialize(operation, term1, term2)

@operation=operation

@term1=term1

@term2=term2

@client=Savon.client(wsdl:"http://localhost:3000/calculator\_soap/wsdl")

elegirOperacion()

end

def elegirOperacion()

case @operation

when "suma"

puts sumar(@term1,@term2)

else

puts "La operacion no es correcta, elige: suma"

end

end

def sumar(term1, term2)

response=@client.call(:add, message: {a:term1, b:term2})

response.body[:add\_response][:value]

end

end

operation=ARGV[0]

term1=ARGV[1]

term2=ARGV[2]

cliente=Client.new(operation, term1, term2)

1. Para probar que funciona correctamente, acceder mediante línea de comando al directorio “*TutorialRuby/client*” y ejecutar el comando “ruby nombreficherocliente.rb operación term1 term2”. Ejemplo para realizar la suma:

cd C:\TutorialRuby\SoapClient

ruby soap\_client.rb suma 1 1